Video Vibrometry for use in Non-Destructive Testing
University of Bristol UNDT

September 3, 2018

Authors:
Alvin Chan

Supervisors:
Prof. Paul Wilcox
Dr. Anthony Croxford

Abstract

Non-Destructive Testing presents many opportunities for in-service assessment of the integrity of parts. The concept of vibrometry through high speed video capture is explored, working on the basis that in-service damage alters vibration modes in ways imperceptible to the human eye, but are made visible through image processing. The validity of the experiment is first considered, recording a simple cantilever beam and extracting Fourier Transforms from the data sets acquired, followed by cross-referencing the values with those obtained by analytical means. SNR curves for a system with known excitation are desired by measuring variables such as input shaker voltage and video duration. With trends established, extrapolating the curves to fit with small spatial amplitudes of excitation leaves the research open to use in the ultrasonic field, possibly making use of stroboscopic effects when excitation frequency exceeds the Nyquist frequency.

1 Introduction

With the advances in CMOS technology over the past few years, they are currently the most viable option for Global Shutter capture, as opposed to the traditional CCD. Though previously plagued by rolling shutter warping, the rapid image acquisition and lack of single-channel ADC bottleneck resulted in CMOS also becoming the most cost effective option when taking into account the reduced cost of production. This also helps to explain their ubiquity, as they are used in places like smartphone cameras.

We wish to take advantage of these modern increases in CMOS processing speeds and resolutions by applying them in a high-speed condition to measure small vibrations in a subject. Though specialised products currently exist on the market capable of performing this task, the aim is to use a comparatively low-cost camera (some two magnitudes cheaper) to obtain the mode shapes and responses for known bodies. In addition, we also aim to exploit short exposure times and the periodic excitations common in UNDT to relate the response shapes to the input signal, even when the excitation frequencies exceed the Nyquist value. These cases are intended to be analysed by taking into account stroboscopic effects produced, as expected natural frequencies and input excitations are either calculable or known respectively.

This presents a hands-off approach to UNDT which makes use of an entire video's worth of data as opposed to a very specific point such as with laser vibrometry. However, the mode shapes and response trends for the Fourier Transform are not well established for this relatively young field. This paper aims to deduce some basic SNR trends with alteration of some of the most common recording parameters, such as input shaker voltage and video duration.

2 Background

The work of N.Wadhwa and A.Davis talk about the usage of video for amplifying small motions and empirically obtaining material properties for fabrics displaced by sound respectively.

2.1 Phase Based Video Motion Processing, N. Wadhwa

The paper describes the use of Fourier phase decomposition, then amplifying the frame-by-frame phase differences to make imperceptible motions visible. By extension, if the code is capable of amplifying miniature oscillations and motions in video, then it must must also be capable of detecting them. Running MIT's source code on our own videos allowed us to extract key data for comparison with the code developed in-house; especially the phase distributions of individual frames. However, their approach to video analysis is fundamentally different to ours and uses an analysis technique known as a "Steerable Pyramid", first developed by Eero Simoncelli in the late 90's. Though its usage is not explored in this paper, its noise attenuating properties warrants investigation in the future, as scaled-colour images showing $d\phi$. (whereas our current code has no such processing for attenuation.)

2.2 Visual Vibrometry: Estimating Material Properties from Small Motions in Video, A. Davis

Though it covers the same concept of video vibrometry, the focus of Davis' work is to derive determine material properties such as elastic modulus for two experimental setups: a vertically cantilevered metal rod and a sheet of hanging fabric. However, with a high performance Phantom camera being used for their metal rod testing (where significant vibration modes only of the order 10^2, their sampling frequency is far below Nyquist. Furthermore, their intended use for vibrometry differs from ours, in that they attempt to deduce material properties as opposed to using manufacturer data and a known input to detect response discrepancies between a safe, undamaged part and the part to be tested.
Nevertheless, Davis' work sets an excellent standard to aspire to as it also makes use of complex steerable pyramids to extract multiple resolutions of phase change data. Its nature as a published paper also makes it far more comprehensive in the history of video vibrometry and overall rigour in testing.

3 Experimentation Procedures and Progression

3.1 Explanation of In-House Code

A code was developed to perform DFTs on a user-specified movie file. It can accept user inputs for a majority of working variables, such as the pixel space of the movie to be used, the size of the analysis kernel used to break down the video, etc. It also features debugging plots which update for every new kernel position when processing a video. This allows users to note properties such as mask size for the kernel, a visualisation of the best-fit velocity plane (especially when said plane has a large gradient) to aid in refining the code, or to allow the user to pause processing to have a more in-depth look at the kernel extracted. The best-fit velocity plane is obtained by a linear regression model which must intersect the origin.

Most importantly, its analysis method is based on calculating the frame-by-frame differences in the video's k-space phase (saved as variable $d\phi$). This technique has been used to great effect with the work of Davis and Wadhwa as above. However, only pixels undergoing significant alterations in time are considered, by defining minimum threshold values. Pixels are only considered for analysis if the phase change exceeds this value. The first build also included circular masking as part of the kernel, but was ultimately omitted owing to its tendency to produce singular matrices which break all further analysis. It also assumes that each change for a pixel is defined as being a single linear transformation. By recalling the complex exponential nature of a Fourier Transform, the 'signal' may be split into an amplitude component and a phase component, where they are independent.

The dy_{2d} and dx_{2d} variables, obtained through gradient measurement for the 2D scalar field of $d\phi$, are among the most useful during analysis; returning velocities and can be Fourier Transformed to give the power spectrum of frequencies present in the video. It is this Fourier Transform which is ultimately used when looking at the Signal-Noise ratio during testing.

3.1.1 `video_processor_parent`

This is responsible for setting up variables and executing everything in one go. It uses MATLAB's native VideoReader capabilities to produce arrays and structured variables for later use (such as extracting the duration of video, which impacts the FFT peaks), although the variable `spatial_downsample` is useful when high-resolution results are not needed, such as when confirming a change in the code or a quick run suffices.

Most importantly, it includes the main loop which applies the analysis to the video's frame space. If the debug axes are enabled, a red box can be seen moving up a column of pixels until shifting the column over to the right, during which a 3D array of N x N x M pixels is sampled for each red box location. N sets the size of the kernel in pixels and M is the total number of frames in the video. (While rectangular kernel bounds are possible, they have a tendency to produce singular matrices during phase analysis and are thus not recommended) This kernel is then translated around, forming a cuboid of data from reading the same location for each frame. Analysis is performed by the function `fn_calc_image_shift_vs_time` and its variants, with `fn_calc_image_shift_vs_time_tester` being the most recent iteration.

The debug axes show a plot for x-direction and y-direction velocities against frame number *(middle plot)*, depictions of a log-scaled heatmap *(top-right plot)*, the dominant feature line overlaid on the same logarithmic heatmap *(middle-right plot)* and a 3D plot of the two planes representing a regression best-fit plane for

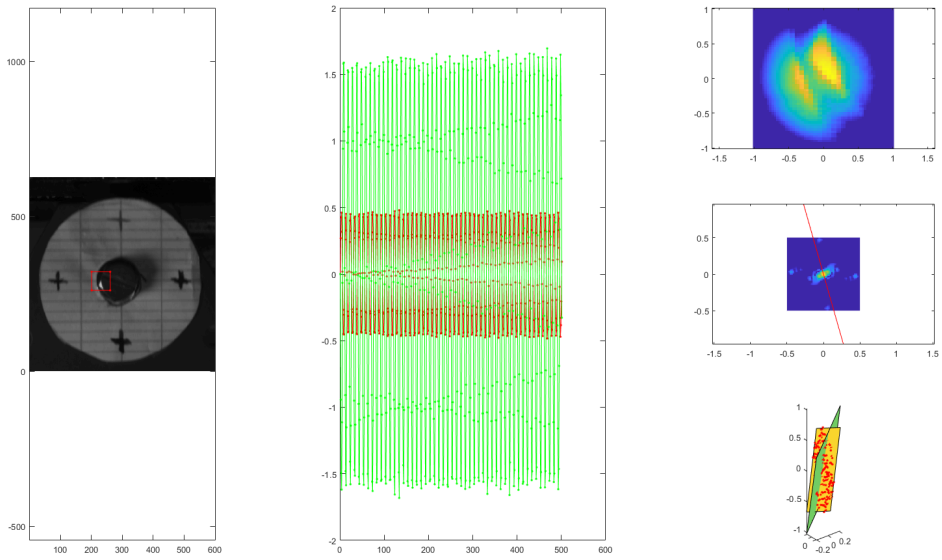

Figure 1: Debug axes example for the kernel passing over a hexagonal nut.

dimensions x, y, and ϕ *(bottom-right plot)*, with an example shown in Figure 1. The definitions of the debug plots are also determined by the sub-function options. The regression plane plotted for the Fourier Domain box also defines how the code obtains the velocity term from the time history of variable $d\phi$ from all frames. `fn_calc_image_shift_vs_time`.

Once the video space has been fully covered by the kernel, it outputs a number of figures. In total, it includes six line plots of different Fourier Transforms obtained from various locations where the kernel was run, a time history plot for a single vertical line of pixels from the video, a 'Co-Ordinate Finder' figure intended for use with MATLAB's Data Cursor, and a greyscale image of the first frame of video. The command window will also return five `NoiseAvg'` values.

In particular, plots such as the FFTs and the `NoiseAvg` values are reliant on manually locating points of interest. Therefore, it is recommended that the locations from which `NoiseAvg` values and FFTs are taken are ascertained after processing to ensure best results (where the FFT and Noise locations are best taken from high-value elements in the Co-ordinate Finder, and the time history plot column of pixels is best found by Data Cursoring the first frame greyscale).

3.1.2 `fn_calc_image_shift_vs_time`

Firstly, the function is intended to return all frame-by-frame Fourier analysis of a kernel. Much like the parent function, it includes various parameters adjustable by the user to alter certain analyses such as whether to return velocity or displacement, although these may be set in the parent function. Its inputs are the 3D array `mov`, which represents kernel data for all frames (unlike its usage in `video_processor_parent` where mov is used to store the entire video data set), and the parameter container `options`. Using the size of the kernel image as a reference, three vectors are defined for x and y directions each (known as x,

kx and kx_shift with corresponding y versions), allowing for a further three matrices to be defined by using `meshgrid`. These vectors and matrices form the basic co-ordinate space which the code works in, with the most common grids in use being `KX_SHIFT` and `KY_SHIFT`. They are also used to define the radial vectors `R`, `KR` and `KR_SHIFT`. Unlike `KX_SHIFT` and `KY_SHIFT`, the radial variants correctly place weighting filters and make use of eigendecomposition.

In short, the function preallocates arrays for the data to come. It also builds the weighting filters as mentioned; favouring the data from every pixel equally would be unwise and prone to noise. Following filter selection and definition, the selected box of movie pixels is first filtered, then Fourier Transformed.

One of the parameters set by `options` is `force_to_dominant_image_feature_direction`, which attempts to determine the locations of sheer edges in the image and orient the directions of calculation to said edges. Setting this value to True allows `Matlab` to take the covariance matrix of the kernel and calculate its corresponding eigenvectors. Since the eigenvector with the largest magnitude also corresponds to the direction of greatest data variation, this is considered to be the normal to the 'edge' under scrutiny. It is also worth noting that MATLAB's way of expressing eigenvectors is to concatenate them as a matrix with each column representing one eigenvector, from left to right in order of decreasing eigenvalues.

Whether the direction is found or not, the Fourier analysis on the array begins, taking $d\phi$ values though division of the k-domain values for a frame and its predecessor. Theoretically, taking the angle of this division returns the change in phase angle, and can be attributed to the exponential nature of a complex number. It is assumed that the amplitude data is irrelevant, since amplitudes in k-domain do not necessarily correspond to the maximum displacement, as opposed to what the spatial amplitude suggests.

Continuing to use this parameter also has effects on subsequent calculations. The angle of the eigenvector is stored in variable `theta` for later use. Before it sees use again, the code needs to determine which elements under its scrutiny should be used when forming a regression plane. The plane is formed in kx space, ky space and using the scalar dϕ as its z-axis. Determining whether a certain point is worth considering is a case of checking that the dϕ value is below a certain threshold (filters out movements which are not considered as part of the vibration.), and the relative amplitude of the point (compared with the largest value in the square at the time).

3.2 Recording Video for Code Analysis

While the code is intended for general use, capture methods are explored to supplement the code and attempt to bypass the Nyquist Frequency limit by exploiting periodicity and the ability to limit exposure by differing phase displacements.

The camera used in experiments was a Blackfly S BFS-U3-32S4C, manufactured by Point Grey FLIR. While not an immensely powerful camera, it features a Global Shutter to prevent image warping as well as a maximum capture rate of 200Hz. FLIR's proprietary software, Spinnaker was also used in conjunction with the camera to ensure compatibility between the computer and the camera.

3.2.1 Hardware Key Specifications

The Blackfly-S used had two connection sources: a USB 3.1 female port (with power supply addition) and a specialised Hirose 6-pin GPIO connection. A summary of their purposes is that the USB cable handles data

transmission between the camera and Spinnaker, while the GPIO connection is used to externally trigger camera captures. While not initially used, the external trigger is to ensure that any strobes implemented will run off the same trigger, preventing small discrepancies in frequency from altering results.
Generation of the external trigger is handled by an Agilent 33220A, outputting a square wave of user-adjustable frequency. Wire is manually soldered on the camera connections and to a Male BNC connection, which allows for a secure connection which is easy to remove. A ground wire camera pin is also connected to the exterior of the non-native BNC connection. Cross-referencing the camera's technical specifications reveals the range of valid voltages for the camera's high and low voltage levels, and should be followed when adjusting the output voltages on the generator.
Since the camera itself is a lone CCD, a Nikon 67mm aspherical lens (code DX SWM ED) is used to focus the image (attached to a corresponding converter for the camera unit.). An improvised solution of blu-tack was used to hold the aperture open since the lens was designed for computer-controlled exposure. It also features two rotary sliders to adjust optical zoom and focus.

3.2.2 Spinnaker Usage

With its installation, Spinnaker can be used standalone or in tandem with MATLAB provided that the correct drivers are installed. However, it is advisable to carry out testing before migrating to MATLAB as it is possible to alter certain video capture parameters during live feedback. Furthermore, the software automatically displays the minimum and maximum limits for parameters when hovered over, giving the user valuable information to with when trying to determine optimal settings to capture vibrations.
Spinnaker possesses the ability to set a time delay between trigger detection and the camera capture proper. This is what allows for alteration of the portion of phase captured. While a majority of the settings available in Spinnaker may be left at their defaults with no serious impact in image quality, the most important section to take note of is the "acquisition control", which adjusts parameters like exposure time, capture mode and input signals.
For this setup, the Hirose connector's 2nd and 5th pins are used, which correspond to an Optically Isolated Input line (henceforth referred to as OPTOIN) and the Optically Isolated Ground line (henceforth referred to as OPTO GND). To correctly set up the trigger of the camera, the Trigger Source in Acquisition must be set to Line 0; the line numbers not matching the pin numbers supplied in the Installation Manual of the camera. Once the required hardware settings have been properly set up, the parameters found in the Features tab must be defined. Referral to the Usage Guide describes which parameters are significant as well as their recommended values in obtaining the data for this report.

4 Experiment Refinement

The original setup held the camera in place by using a clamp and boss combination. While this was sufficient when quick removal and re-positioning was needed, the camera assembly was front heavy. This meant that adjusting the pitch of the camera needed overcompensation as there was a tendency for the clamp rod to slide after releasing it in position. However, the improved setup uses aluminium structural bars and plates screwed onto the camera's mounting screw holes. The assembly is then G-clamped onto a adjustable-height platform.

In addition, the vibration to be recorded is more consistently produced compared to the original. An electromagnetic shaker, attached to a paper target (providing more image contrast for the code to pick up) allows us to rigorously alter the vibration parameters by connecting it to the signal generator as mentioned prior. The paper's matte texture also makes it more consistent when illuminated by DC LEDs. Establishing the code's ability to pick up on small vibrations was determined by varying the actuation voltage or the recording time. From this series of videos, graphical presentations of SNR and FT peaks could be created.

5 Results

5.1 Altering Voltage

Figure 2 shows the SNR curve for large voltages in the range of 0.5V to 10V input to the shaker, while figure 3 refers to voltages between 0.1V and 1.5V. All voltage tests were carried out with an exposure time of 9003μs, and actuation frequency of 10Hz, although the large-voltage videos are 5s long as opposed to the 10s for the small voltages. In both plots, the different colours signify which marker on the target is being considered. In both cases, there is a clear trend which shows increasing SNR with vibration amplitude, although the figure from Volt1 is distinctly non-linear, as opposed to the plot for small voltages. However, it must be noted that these plots are for the FFT harmonic amplitude against the mean noise calculated for the dataset. Notably, the peak for the first harmonic does not exceed the maximum noise until input voltages greater than 0.3V are used.

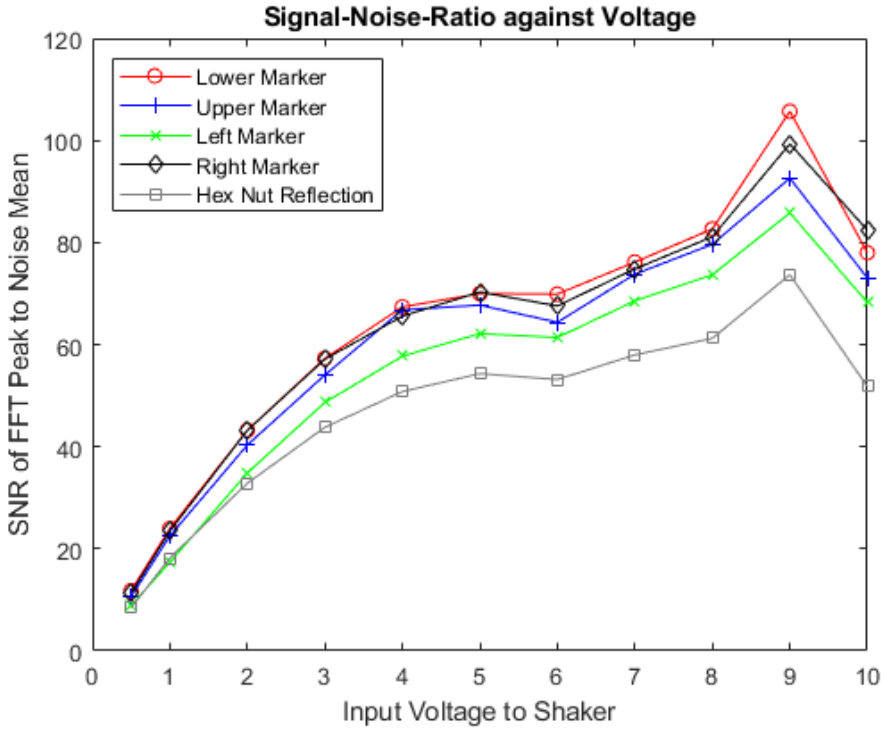

Figure 2: SNR:Voltage for 0.5V to 10V

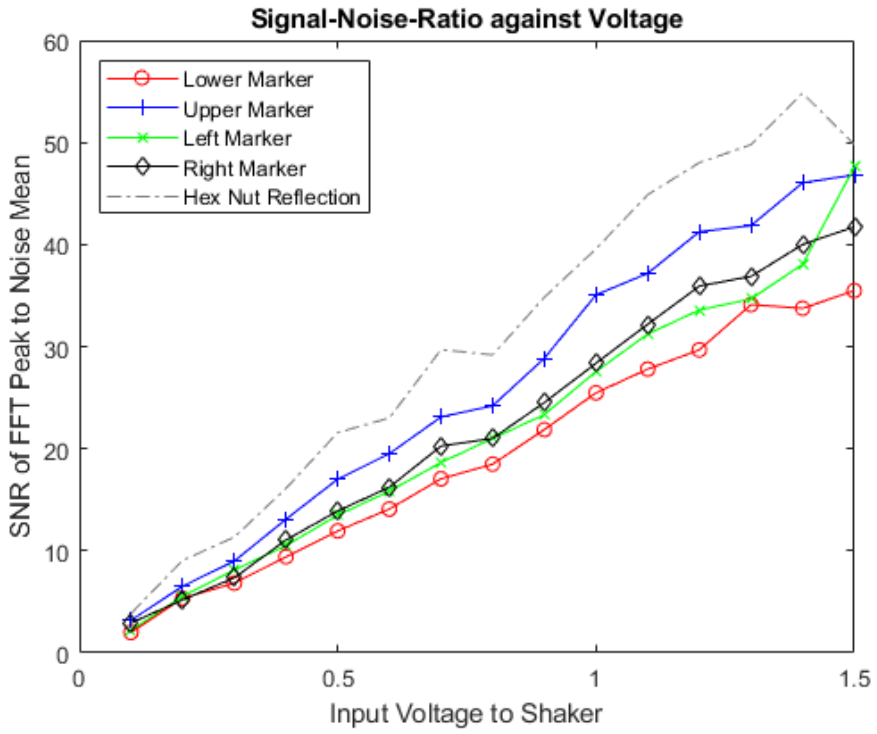

Figure 3: SNR:Voltage for folder Volt2

Table 1 is taken from the right marker's data for low voltages. Since the mean noise also incorporates the minima in the noise, it is reasonable to ensure that maximum noise peaks are less than the desired peaks when deciding on data to draw conclusions from.

V_{pp}	1st Harmonic	NoiseMax	NoiseAvg
0	N/A	7.893	1.6603
0.1	6.207	8.312	2.12
0.2	12.23	3.159	2.3542
0.3	17.76	19.09	2.4161
0.4	25.94	9.907	2.3455
0.5	33.66	9.828	2.4213
0.6	40.53	8.785	2.4981
0.7	48.91	9.092	2.4145
0.8	56.81	19.22	2.7041
0.9	66.55	10.69	2.7143
1	73.66	9.011	2.5965
1.1	81.06	11.75	2.519
1.2	92.05	7.386	2.5618
1.3	100.4	13.15	2.7228
1.4	108.2	9.53	2.7047
1.5	118.6	8.504	2.8413

Table 1: The peak at the first harmonic does not exceed the NoiseMax until 0.4V.

Fortunately, the peak to peak amplitudes correspond relatively well with the input voltage, with the apparent trend being

$$A_{pp} = V_{in} + 1 \tag{1}$$

...with the constant likely originating from the 'central line' existing as a line of pixels. Since the scale of the video, assuming small angles, is roughly 31 pixels to 2mm (based on the line separation of the paper), it may be deduced that, using 0.4V as the cut-off, the minimum peak to peak pixel amplitude required is 0.4 pixels, or 0.026mm for this setup. This estimate assumes a linear relationship and sufficient contrast in the image to pick up (with the best locations to take either being the lower marker or the light reflection on the hexagonal nut). However, minor differences in ambient lighting and exact position of the target in relation to the camera may impact these values (especially since the kernel's translation step was 20 pixels for these cases.)

5.2 Altering Recording Time

The aim was to also determine the extent to which longer recording times benefit SNR. In a manner similar to the voltage testing videos, the SNR and Voltage plots are given for 5V input voltage, 9003μs exposure time and 99.98fps recording rate.

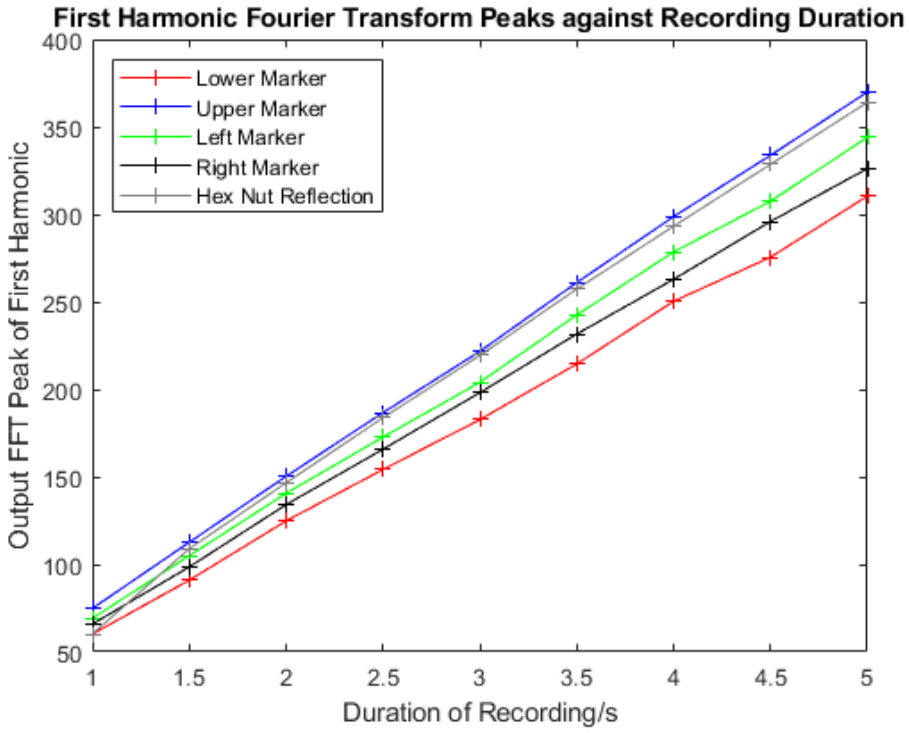

Figure 4: FFT Peak against Recording Time

Though the Fourier Plot peak response to increased recording duration is a clear linear trend, the SNR is not. Running the code on the videos for 3.5s and 4.0s recordings show a significant increase, although the

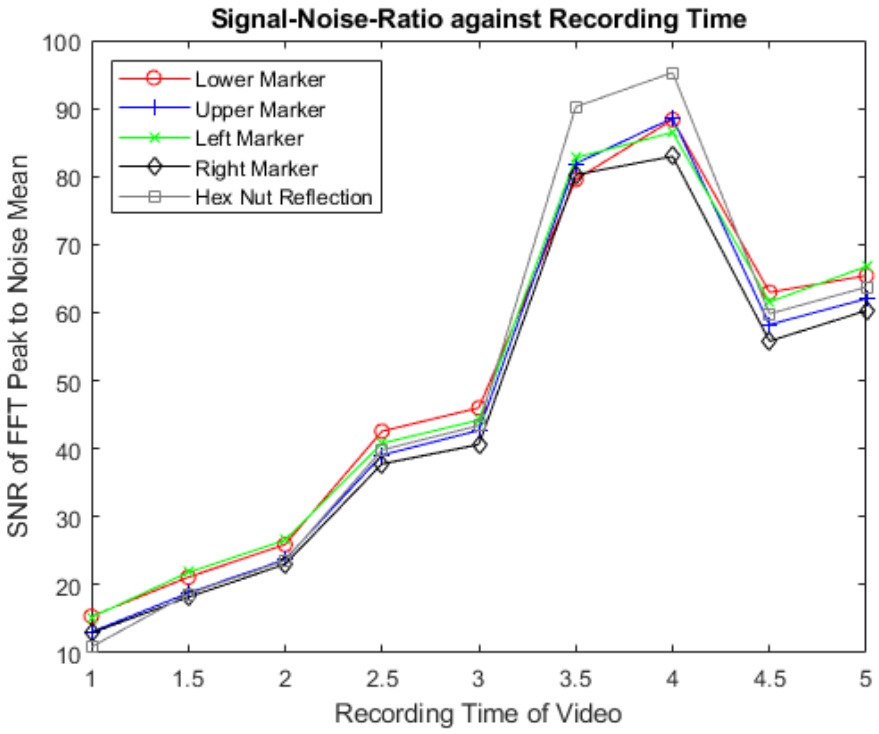

Figure 5: SNR against Recording Time

cause is unknown. Given that the videos having were taken in quick succession, changes in lighting may be discounted as a reason. The presence of the SNR spike in two readings in the middle of the data set also rule out the possibility that there was a small displacement of the equipment to alter the features captured in each kernel.

At best, it is expected that the benefits of increased recording diminish with larger recording times as the Fourier Transform has a natural averaging quality to it; a square rooting or logarithmic decrement is expected. It would not be unreasonable to assume that a maximum SNR value attainable for this set of parameters is 100.

6 Conclusions

Analysing the SNR trends for Voltage was the core focus of this project, as it establishes a value for the smallest pixel resolution recordable by the code. However, it is worth noting how the pixel amplitude was subjectively obtained (albeit with a set procedure of vertically scaling to only look at one sinusoid). Furthermore, it is clear that there is no detriment to data quality when recording for longer periods of time; the main constraints are often the storage space of the machine running SpinView (with a 5 second, 100fps video consuming just under 0.5GB)

Videos for other variables were also recorded, such as changing capture rate or exposure time per frame. However, they have been omitted due to time constraints. One notable result that arose was that a threshold

for minimum exposure time could be established in a future work as the shortest time used was 150μs, and produced significantly more noise compared to its brighter-lit videos.
Plots showing the distribution (and peaks) of noise plots were also created but were purely specific to our system and thus saw little usage outside of being a curiosity. Of note was that the videos recorded for folder Volt2 seemed to have a lot of its noise peaks around frequency.

7 Further Avenues of Research

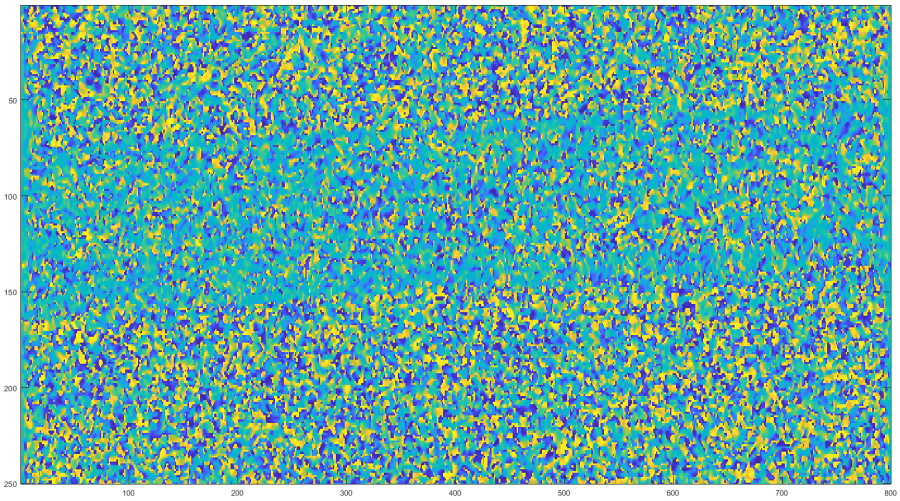

Figure 6: The $d\phi$ plot for layer 2 when processed by MIT's Phase Based amplification code.

Figure 6 illustrates how video frames (and their corresponding phase changes) may be noise attenuated when properly filtered, possibly by wavelets (since the noise seems to have formed clusters of in-phase pixels.). Furthermore, voltage and duration testing may be further pushed to their limits if an amplifier is acquired. This would help to cement the existing trends observed in the graphs thus far, as well as allow the true nature of odd-shaped plots like figure 5 to be found. Other video recordings have been taken and also remain unused, either owing to time-constraints or external factors affecting results, an example being a case where the shaker's vibrations were running through the surface and to the camera.
Since the end goal of this code is to use video vibrometry in a non-destructive testing context, removing noise becomes a high priority. Ultrasonic NDT suggests even smaller actuation amplitudes compared to those of this report, owing to the high frequencies at work. This means that attempting to find characteristic vibration modes when sampling under the Nyquist frequency becomes of greater importance, and will likely require further research into how stroboscopic effects would affect the Discrete Fourier Transforms of such cases.